

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

This Page Blank (uspto)

E 6047

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-109758

(43)Date of publication of application : 20.04.2001

(51)Int.Cl. G06F 17/30
G06F 3/00
G06F 12/00

(21)Application number : 11-285164

(71)Applicant : HITACHI LTD

(22)Date of filing : 06.10.1999

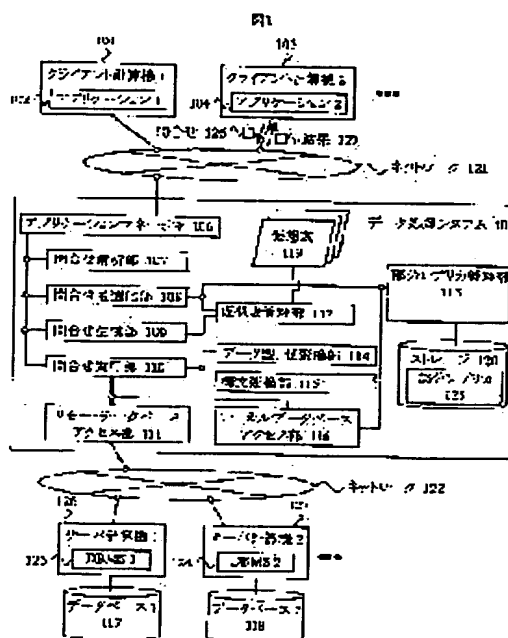
(72)Inventor : NISHIZAWA ITARU
INOHARA SHIGEKAZU
SAGAWA NOBUTOSHI
SHIMIZU AKIRA

(54) VIRTUAL TABLE INTERFACE, AND SYSTEM AND METHOD FOR PROCESSING INQUIRY USING THIS INTERFACE

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a data processing system that is a foundation to construct an information system capable of flexibly corresponding to a change of access object data and further to provide a mechanism which improves the performance of the data processing system needing distributed inquiry processing.

SOLUTION: The data processing system which receives an inquiry from an application is provided internally with a virtual table capable of multiple mapping to a plurality of databases (1). A partial replica being the copy of a part of data on the virtual table is prepared, and all or a part of the inquiry is processed by using the partial replica (2). Also, about an inquiry unit in which the partial replica can not be used, appropriate distributed join system selection and optimization that pushes down processing to a real database as much as possible are used.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

This Page Blank (uspto)

E6047

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-109758

(P2001-109758A)

(43) 公開日 平成13年4月20日 (2001.4.20)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード [*] (参考)
G 0 6 F 17/30		G 0 6 F 3/00	6 5 6 C 5 B 0 7 5
3/00	6 5 6	12/00	5 1 3 D 5 B 0 8 2
12/00	5 1 3	15/40	3 8 0 D 5 E 5 0 1
			3 1 0 F
			3 1 0 C

審査請求 未請求 請求項の数21 O L (全 17 頁)

(21) 出願番号 特願平11-285164

(22) 出願日 平成11年10月6日 (1999.10.6)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 西澤 格

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(72) 発明者 猪原 茂和

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(74) 代理人 100075096

弁理士 作田 康夫

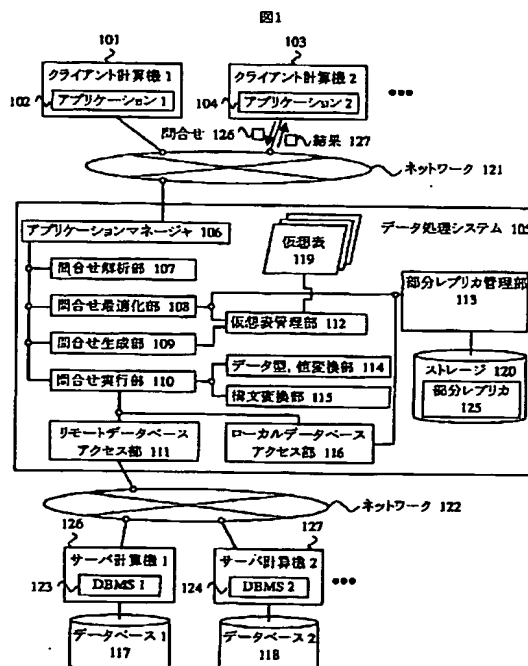
最終頁に続く

(54) 【発明の名称】 仮想表インタフェースと該インタフェースを用いた問合せ処理システム及び方法

(57) 【要約】 (修正有)

【課題】 (1) アクセス対象データの変更に柔軟に対応可能な情報システムを構築する基盤となるデータ処理システムを提供する。さらに、(2) 分散問合せ処理を必要とする前記データ処理システムの性能を向上する機構を提供する。

【解決手段】 (1) アプリケーションからの問合せを受け付けるデータ処理システム内に、複数データベースへの多重マッピングが可能な仮想表を設ける。(2) 仮想表上の一部のデータのコピーである部分レプリカを作成し、問合せの全部もしくは一部を部分レプリカを利用して処理する。また、部分レプリカを利用できない問合せ単位については、適切な分散ジョイン方式の選択と、処理を可能な限り実データベースにブッシュダウンする最適化を用いる。



【特許請求の範囲】

【請求項1】 ユーザアプリケーションから実データベースを隠蔽する仮想表を用いて実データベースへマッピングを行うことにより実データベース上の項目をアクセスして問合せを実行するための仮想表インタフェースであって、1個の仮想表に複数の互いに異なるマッピングを割り当て、アクセス条件に応じて該マッピングの切り替えを行うことを特徴とする仮想表インタフェース。

【請求項2】 アプリケーションを実行する少なくとも一つのクライアントと、それぞれにデータベースを備え、アクセス要求にしたがって該データベースを検索する少なくとも一つのサーバとにネットワークで接続され、前記クライアントからの問合せを受け付け、該問合せに対する結果を該クライアントに返す問合せ処理システムであって、前記少なくとも一つのデータベース上の項目、もしくは前記項目に対する演算結果の多重マッピングが可能な仮想項目名を保持して前記アプリケーションに提供する仮想表と、該仮想表を管理する仮想表管理部と、前記問合せを解析する問合せ解析部と、該問合せ解析部で解析された問合せから、該仮想表の定義を用いて前記データベースを参照するデータベース参照問合せを生成する問合せ生成部と、該問合せ生成部で生成された前記データベースへの問合せ処理を実行する問合せ実行部と、を有することを特徴とする問合せ処理システム。

【請求項3】 アプリケーションを実行する少なくとも一つのクライアントと、それぞれにデータベースを備え、アクセス要求にしたがって該データベースを検索する少なくとも一つのサーバとにネットワークで接続され、前記クライアントからの問合せを受け付け、該問合せに対する結果を該クライアントに返す問合せ処理システムであって、前記少なくとも一つのデータベース上の項目、もしくは前記項目に対する演算結果のマッピング、もしくは多重マッピングが可能な仮想項目名を保持して前記アプリケーションに提供する仮想表と、該仮想表を管理する仮想表管理部と、該仮想表に定義された項目、もしくは前記項目に対する演算結果の全体もしくは一部分を実データとして保持する部分レプリカと、該部分レプリカ内のデータを管理し、データの検索、データ更新、データ削除、データ挿入を行う部分レプリカ管理部と、前記問合せを解析する問合せ解析部と、該問合せ解析部で解析された問合せが、前記部分レプリカを用いて処理可能か否かを判定する問合せ最適化部と、処理可能と判定された場合に、該問合せから該部分レプリカを参照する部分レプリカ参照問合せを生成する部分レプリカ参照問合せ生成部と、処理不可能と判定された場合に、該問合せから前記仮想表の定義を用いて前記データベースを参照するデータベース参照問合せを生成するデータベース参照問合せ生成部と、生成された前記部分レプリカ参照問合せか前記データベース参照問合せの少なくとも一方を実行する問合せ実行部と、を有することを特徴とする

問合せ処理システム。

【請求項4】 前記部分レプリカ管理部は、汎用のデータベース管理システムを有する請求項3記載の問合せ処理システム。

【請求項5】 前記仮想表管理部は、多重マッピングされている仮想表を、データ鮮度、データ精度、ユーザロール、業務、アクセス位置のいずれか、もしくはその組合せを指定して切り替える手段を有する請求項2記載の問合せ処理システム。

【請求項6】 前記問合せ最適化部は、前記解析された問合せを選言標準形に変形する手段と、該選言標準形の各要素に対して前記部分レプリカを用いて処理可能か否かを判定する手段と、処理可能と判定された場合には前記選言標準形の要素に対して、該要素の処理に必要な部分レプリカの組を生成する手段と、を有する請求項3記載の問合せ処理システム。

【請求項7】 アプリケーションを実行する少なくとも一つのクライアントと、それぞれにデータベースを備え、アクセス要求にしたがって該データベースを検索する少なくとも一つのサーバとにネットワークで接続され、前記クライアントからの問合せを受け付け、該問合せに対する結果を該クライアントに返す問合せ処理システムであって、前記少なくとも一つのデータベース上の項目、もしくは前記項目に対する演算結果のマッピングもしくは多重マッピングが可能な仮想項目名を保持して前記アプリケーションに提供する仮想表と、該仮想表を管理する仮想表管理部と、前記問合せを解析する問合せ解析部と、該問合せ解析部で解析された問合せから、該仮想表の定義を用いて前記データベースを参照するデータベース参照問合せを生成する問合せ生成部と、該問合せ生成部で生成された前記データベースへの問合せ処理を実行する問合せ実行部とを有し、前記データベース参照問合せを生成する問合せ生成部は、問合せを選言標準形に変形する手段と、該選言標準形の各要素である問合せ要素に対して、該問合せ要素の処理が一つのデータベースで処理可能か否かを判定する手段と、処理可能と判定された場合には、該問合せを該データベースで処理するように書き替える手段と、処理不可能と判定された場合には、該問合せを該データベースとデータ処理システムとを用いて処理するように書き替える手段と、を有する問合せ処理システム。

【請求項8】 前記問合せをデータベースとデータ処理システムとを用いて処理するように書き替える手段は、該問合せが異なるデータベース間の結合処理を含む問合せに対して、以下の方式(1)～(4)それぞれの適用可能性をチェックする手段と、(1)結合対象の第1のデータベースおよび第2のデータベースからそれぞれ結合対象のデータをデータ処理システムに転送し、該データ処理システムで結合処理を行う第1の方式、(2)第1のデータベースから得た結果を第2のデータベース検索

の問合せに埋め込むことによって結合処理を行う第2の方式、(3)第2のデータベースに対する検索式に検索条件をパラメータとして記述し、第1のデータベースから得た結果をパラメータの値として与えて結合処理を行う第3の方式、および(4)第1のデータベースから得た値を第2のデータベースの一時表に挿入し、第2のデータベース上で結合処理を行う第4の方式、適用可能な方式の中から、処理コスト計算に基づき最もコストが小さい方式を選択する手段と、を有する請求項7記載の問合せ処理システム。

【請求項9】前記問合せ生成部は、問合せをデータベースとデータ処理システムの両方を用いて処理するように書き替える手段と、複数の問合せ実行方法の候補を抽出する手段と、異なるデータベース間の複数の結合処理を含む問合せにおいて、抽出された複数の候補間の結合処理に対して、以下の方式(1)～(4)それぞれの適用可能性をチェックする手段と、(1)結合対象の第1のデータベースおよび第2のデータベースからそれぞれ結合対象のデータをデータ処理システムに転送し、該データ処理システムで結合処理を行う第1の方式、(2)第1のデータベースから得た結果を第2のデータベース検索の問合せに埋め込むことによって結合処理を行う第2の方式、(3)第2のデータベースに対する検索式に検索条件をパラメータとして記述し、第1のデータベースから得た結果をパラメータの値として与えて結合処理を行う第3の方式、および(4)第1のデータベースから得た値を第2のデータベースの一時表に挿入し、第2のデータベース上で結合処理を行う第4の方式、前記問合せ実行方法の候補から適用不可能な方式を含む問合せ実行方法の候補を取り除いた残余の各候補の処理コストを計算する手段と、該処理コストが最も小さい候補を問合せ実行方法として選択する手段と、を有する請求項7記載の問合せ処理システム。

【請求項10】前記処理コストを計算する手段は、計算処理、I/O処理に必要なCPU時間の和で見積もる手段を有する請求項8または9記載の問合せ処理システム。

【請求項11】前記処理コストを計算する手段は、該コストをデータ転送量で見積もる手段を有する請求項8または9記載の問合せ処理システム。

【請求項12】前記問合せ実行部は、データベース参照問合せを実行する際に問合せ構文を変換する構文変換手段と、問合せ条件のデータ型および値と、該変換された問合せの結果、前記データベースより得られるデータのデータ型および値を変換するデータ型、値変換手段と、を有する請求項2または3記載の問合せ処理システム。

【請求項13】前記データ型、値変換手段は、データ型識別子とデータ本体の組で、データベース固有のデータを格納する普遍データ型を持つことによりデータの精度を落さずにデータの格納およびデータ変換を行う手段を

有する請求項12記載の問合せ処理システム。

【請求項14】前記構文変換手段、およびデータ型、値変換手段は、それぞれ、変換に必要なモジュールをプラグインとして動的に挿入可能なインタフェース手段を有する請求項12記載の問合せ処理システム。

【請求項15】前記仮想表管理部は、前記仮想表の定義を行うインタフェース手段を有する請求項2記載の問合せ処理システム。

【請求項16】前記仮想表管理部は、前記仮想表に対する多重マッピングの切り替えを行うインタフェース手段を有する請求項2記載の問合せ処理システム。

【請求項17】前記部分レプリカ管理部は、前記部分レプリカの定義を行うインタフェース手段を有する請求項3記載の問合せ処理システム。

【請求項18】アプリケーションを実行する少なくとも一つのクライアントと、それぞれにデータベースを備え、アクセス要求にしたがって該データベースを検索する少なくとも一つのサーバとにネットワークで接続されたデータ処理システムにおいて実行されるべき問合せを前記クライアントから受け付け、該問合せの実行結果を該クライアントに返す問合せ処理方法であって、受け付けた問合せを解析する処理と、前記少なくとも一つのデータベース上の項目、もしくは前記項目に対する演算結果の多重マッピングが可能な仮想項目名を保持して前記アプリケーションに提供する仮想表の定義を用いて、該解析された問合せから前記データベースを参照するデータベース参照問合せを生成する処理と、生成した問合せを実行する処理と、を有することを特徴とする問合せ処理方法。

【請求項19】アプリケーションを実行する少なくとも一つのクライアントと、それぞれにデータベースを備え、アクセス要求にしたがって該データベースを検索する少なくとも一つのサーバとにネットワークで接続されたデータ処理システムにおいて実行されるべき問合せを前記クライアントから受け付け、該問合せの実行結果を該クライアントに返す問合せ処理方法であって、受け付けた問合せを解析する処理と、前記少なくとも一つのデータベース上の項目、もしくは前記項目に対する演算結果の多重マッピングが可能な仮想項目名を保持して前記アプリケーションに提供する仮想表の定義を用いて、該解析された問合せから前記データベースを参照するデータベース参照問合せを生成する処理と、該仮想表に定義された項目、もしくは前記項目に対する演算結果の全体もしくは一部分を実データとして保持する部分レプリカを参照する部分レプリカ参照問合せを生成する処理と、該部分レプリカを用いて、前記解析された問合せが処理可能か否かを判定する処理と、処理可能と判定された場合には、少なくとも前記部分レプリカ参照問合せを実行する処理と、処理不可能と判定された場合には、前記データベース参照問合せを実行する処理と、を有すること

を特徴とする問合せ処理方法。

【請求項20】アプリケーションを実行する少なくとも一つのクライアントと、それぞれにデータベースを備え、アクセス要求にしたがって該データベースを検索する少なくとも一つのサーバとにネットワークで接続されたデータ処理システムにおいて実行されるべき問合せを前記クライアントから受け付け、該問合せの実行結果を該クライアントに返す問合せ処理方法であって、前記少なくとも一つのデータベース上の項目、もしくは前記項目に対する演算結果のマッピングもしくは多重マッピングが可能な仮想項目名を保持して前記アプリケーションに提供する仮想表の定義を用いて前記データベースを参照するデータベース参照問合せを生成する処理と、該生成されたデータベース参照問合せを実行する処理とを有し、前記データベース参照問合せを生成する処理は、生成された問合せを連言標準形に変形する処理と、該連言標準形の各要素である問合せ要素に対して、該問合せ要素の処理が一つのデータベースで処理可能か否かを判定する処理と、処理可能と判定された場合には、該問合せを該データベースで処理するように書き替える処理と、処理不可能と判定された場合には、該問合せを該データベースとデータ処理システムとを用いて処理するように書き替える処理と、を有することを特徴とする問合せ処理方法。

【請求項21】請求項18～20いずれか1項記載の方法をコンピュータを用いて実行するためのコンピュータプログラムを記録した記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、データ処理に関し、特に、複数データベースへの統合アクセスを実現する問合せ処理システム及び方法に関する。

【0002】

【従来の技術】企業内情報システムの再編、企業間提携をはじめとする社会情勢の急速な変化に対応可能な情報システムが求められるようになってきている。通常企業内には多数のデータベースが存在し、それぞれのデータベースは膨大なデータを多数のファイルやテーブルに保持している。これらのデータは長い時間をかけて様々な状況下で作成されてきたという経緯から一貫性が無く、

(1) 別々のデータにアクセスするためには個別のアプリケーションを使用せざるを得ない、(2) 新業務の開始、業務内容変更にあたっては新規アプリケーション開発、アプリケーション変更を行わなければならないという問題が指摘されていた。個別データにアクセスするために個別アプリケーションを利用するアプローチは、アプリケーションの数および種類が多くなることに起因する管理の複雑化、多数のアプリケーションの開発および維持にかかるコスト増加、そしてアプリケーション開発に必要となる時間分の業務遅延に起因する非効率化等の

数多くの欠点を持つのは明らかである。

【0003】アプリケーションから複数のデータベースを隠蔽する目的で仮想的なテーブル（以下、仮想表と呼ぶ）を作成して該仮想表上の項目から実在するデータベース（以下、実データベースと呼ぶ）上の項目へのマッピングを用いることによって、複数の実データベースへの透過アクセスを実現する方式が、(1) 米国特許5873088 “Derived data base processingsystem enabling one program to access a plurality of data basis”（文献1）や、(2) 米国特許5675785 “Data warehouse which is accessed by a user using a schema of virtual tables”（文献2）に開示されている。(1)は実データベースの論理定義情報と格納情報を用いて複数の実データベースに透過的にアクセスする方式、(2)は仮想テーブルで構成されるスキーマに対して発行された問合せを変換して実データベースにアクセスする方式で、共に実データベースを隠蔽し、アプリケーションから仮想的な表へのアクセスを実データベースへのアクセスに変換することを特徴としていた。仮想的なスキーマを介して実データベースにアクセスするという考え方は、データベース統合やスキーマ統合と呼ばれ、1980年頃より学会を中心として研究が盛んに行われ、例えばA. Sheth, J. Larson著、“Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases”、ACM Computer Surveys, Vol. 22, No. 3, pp. 183-236（文献3）に示されている連邦データベースシステムに代表されるいくつかの統合方式が提案されている。これらの方式はいずれも仮想的なスキーマもしくは表から実データベースへのマッピングを用いて、ユーザもしくはアプリケーションから実データベースを隠蔽し、論理的な統合を実現していたが、一つの仮想表に対して異なる複数のマッピングを割り当て、該複数マッピングをアクセス要求の条件に応じて切り替えるという考え方は従来技術にはなかった。その理由は、従来技術では実体が異なる複数のデータを一つの仮想表に割り当てる利点が明確でなく、割り当てたマッピングを切り替えるための基準、機構の検討が行われていなかったためである。最近、情報システムの複雑化、多様化に対応するために仮想表を複数のアプリケーションで共有し、さらにアプリケーションからの仮想表のアクセス条件に応じて異なる実データベース上の項目をアクセス可能にしたいという要求が高まってきた。しかし、前記マッピングを切り替える機構をアプリケーションで実現するという従来の単純な方法では該ア

ブ리케이션の構造が複雑化するという問題があった。

【0004】仮想表経由の透過アクセスを利用した大規模な企業情報システムを構築する際のもう一つの問題点は、業務を遂行するための現実的な性能が得られないということである。特に複数のデータベースやデータウェアハウスに対して分散問合せを行う環境で、OLAP (OnLine Analytical Processing) に代表される複雑な問合せを実行する場合に、その問題が顕著になる。例えばデータ規模に関しては、1999年7月現在で数TB (テラバイト; 10の12乗バイト) 級のデータウェアハウスが構築されており、米国の先進的な企業では投入から答えを得るまでに1日以上を有する複雑な問合せを発行するユーザが現れつつあることが報告されている。このような複雑な問合せは、多数の大規模な表の結合処理などの負荷の高い処理を含んでいる。結合処理はアドホックな解析系の問合せで多発する表の付き合わせ処理であるが、処理対象となる表が異なるデータベース上に存在する場合 (以下、分散問合せ処理と呼ぶ) にはデータベース間でのデータ転送が発生するため効率は極端に悪くなる。

【0005】分散問合せ処理の効率をあげるための方式としては、(1) 問合せの最適化を行うことによって、個別の実データベースで可能な処理を個々の実データベースにプッシュダウンすることによってデータ転送量と処理量を削減する方式、(2) 処理対象データをキャッシュし、該キャッシュデータを利用してデータ転送を省略し、処理を高速化する方式が考えられる。例えば

(1) の処理のプッシュダウンについては、米国特許5590321 "Push down optimization in a distributed, multi-database system" (文献4) にその方式が開示されている。本方式は、問合せが処理に必要とするデータ、および機構を保持する実データベースに処理をプッシュダウンするが、処理のプッシュダウンの単位を問合せ全体、もしくは副問合せとしているために、前述のOLAPに代表される複雑な問合せのように、さらに細かな問合せ単位でなければプッシュダウンが可能でしかも効果が大きい場合に対して適用できないというように適用範囲が限定されてしまうという問題がある。異なるデータベース間での結合処理 (以下、分散ジョインと呼ぶ) を実行する際には、該結合処理をどこでどのような方式で実行するかによって、データ転送量やデータベース内部処理量などシステム全体に影響を及ぼす処理量は大きく変化する。ところが、従来は前記データ転送量や前記データベース内部処理量を考慮して、前記分散ジョインの実行方式と実行場所を適切に選択することによって処理量全体を最小化するという方法は用いられていなかった。

【0006】(2) のデータのキャッシュに関しては、

従来技術として (i) 従来の計算機システムに実装されているキャッシュメモリ、(ii) 近年研究が盛んなWEBキャッシュ、そして (iii) データベース向けに開発されたキャッシュがあげられる。(i) ではキャッシュメモリ上にアドレスとデータの組が格納され、あるアドレスのデータが要求された場合に、該アドレスがキャッシュ上にある時には該アドレスに対応するデータがキャッシュから返される。また、(ii) ではキャッシュ上にWEBデータを唯一に識別するアドレスであるURL (Universal Resource Locator) とデータの組が格納され、あるURLが要求された場合に、該URLがキャッシュ上にある時には該URLに対応するデータがキャッシュから返される。つまり、(i) および (ii) ではデータを唯一識別可能なアドレスが与えられ、該アドレスがキャッシュ上に存在するか否かという判定しか行わず、キャッシュされたデータそのものが利用可能か否かを判定していた。そのため、キャッシュしたデータの一部分が他のリクエストで利用可能な場合のデータの再利用について考慮していなかった。ところが、前述のOLAPに代表される大量データに対する解析系の処理では対象となる大規模データに対し、少しずつ条件を変えながら様々な軸で解析を行うため、前回と全く同一の問合せを発行することは稀であり、前記 (i) もしくは (ii) の従来方式の適用は困難であった。

【0007】(iii) のデータベース向けに開発されたキャッシュで、キャッシュデータを利用して無駄なデータ転送を省略する方式として、例えば米国特許5903887 "Method and apparatus for caching result sets from queries to a remote database in a heterogeneous database system" (文献5) に、結合処理時に結合の重複率が高い場合に結合対象データをキャッシュして処理の効率を上げる方式が開示されているが、開示方式は適用範囲が動的な結合処理に限定されてしまい、一般の問合せに適用が難しいという問題があった。また、A. Keller, J. Basu 著, "A Predicate-based Caching Scheme for Client-Server Database Architectures", The VLDB Journal, Vol. 5, No. 1, pp. 35-47 (文献6) では、問合せと該問合せ処理結果データをキャッシュし、新たな問合せを該キャッシュデータを利用して処理することにより、サーバ負荷の削減と問合せ処理時間の短縮を行う方式が提案されている。本方式は問合せ結果の再利用率が高い場合にはサーバ負荷削減、問合せ処理時間短縮に大きな効果があるが、大規模な複数のデータベースに接続される環境を想定した場合には、キャッシュ対象となるデータ

規模とデータ処理システム内で利用可能なキャッシュ用の記憶装置の規模の比が極めて大きいため、キャッシュされたデータの再利用率が極めて低くなりシステム全体の効率が悪くなってしまうという欠点がある。

【0008】

【発明が解決しようとする課題】企業活動に必要なデータを蓄積するデータベースが多数存在する環境では、アクセス対象となるデータ毎にアプリケーションを切り替えながらデータベースを利用する必要があり、従来よりアプリケーションの作成、管理、維持にかかる時間、コストが増加する問題があった。また、複数のデータベースを利用する分散問合せ処理で、業務遂行に必要な現実的な性能が得られない問題もあった。

【0009】従って、本発明の第1の目的は、アクセス対象のデータの変更に柔軟に対応可能な情報システムを構築する基盤となる問合せ処理システム及び方法を提供することである。さらに、本発明の第2の目的は、分散問合せ処理を必要とする問合せ処理の性能を向上する機構を提供することである。

【0010】

【課題を解決するための手段】前記第1の目的を達成するため、本発明ではアプリケーションからの問合せを受け付けるデータ処理システム内に、複数データベースへの多重マッピングが可能な仮想表を設ける。仮想表上の項目は実データベース上の表の項目、もしくは実データベース上のビューの項目、もしくは他の仮想表の項目（以下、単にデータベース上の項目と呼ぶ）、もしくはこれらの項目に対する演算結果にマッピングされ、アプリケーションからは仮想表上の項目を参照し、データベース上の項目に対してではなく仮想表上の項目に対して問合せを発行する。これにより、アプリケーションから複数データベースへのアクセスを隠蔽できる。本発明のデータ処理システムにおける多重マッピングとは、一つの仮想表が異なる条件の複数のマッピングを保持できることを意味する。

【0011】多重マッピング機構を用いることにより、アプリケーションは仮想表の切り替え、仮想表上のアクセス項目の変更無しにアクセス先データベースを切り替えることが可能となる。アクセス先を切り替えながら複数のデータベースを取り扱う際には、データベースシステム固有のデータ型を取り扱う必要があるが、本発明のデータ処理システムではある実データベースのみでサポートされているデータ型を取り扱うための普遍データ型を導入し、データ処理システムから該データ型に対する操作を可能とする。以上の機構により、アクセス対象のデータの変更に柔軟に対応可能な情報システム構築基盤を提供できる。

【0012】前記第2の目的を達成するため、本発明のデータ処理システムでは仮想表上の項目、もしくは前記項目に対する演算結果の全体もしくは一部分を、問合せ

処理に先立ち実データとしてデータ処理システム内のストレージ、もしくはデータ処理システムに高速ネットワークで接続されたストレージに格納する。前記格納された実データを部分レプリカと呼ぶ。多数の仮想表が存在する場合に、該仮想表に対する部分レプリカがデータ処理システム内に占めるデータ量を削減するために、部分レプリカは多数の仮想表から共有可能とする。前記データ処理システムでは、アプリケーションからの仮想表に対する問合せは論理的に等価なより細かい単位に分解し、それぞれの単位で部分レプリカを利用可能か否かを判定する。部分レプリカが利用可能な問合せの単位に関しては、部分レプリカを利用して処理することによってデータ転送量を削減し、性能を向上させる。また、部分レプリカを利用できない問合せ単位については、分解した問合せを論理的に等価な別の形にさらに変換し、変換後の問合せ単位毎に該問合せが一つの実データベースで処理可能か否かを判定し、一つの実データベースだけを用いて処理できる問合せは該処理を該実データベースにブッシュダウンする最適化を用いることによってデータ転送量を削減し、性能を向上させる。さらに、システムへの負荷が高い複数の実データベース間の結合処理が発生する場合には、転送されるデータ量に応じて複数の分散ジョイン方式を適切に切り替えることによって、データ転送量およびデータベース内部処理量を削減し、性能を向上させる。

【0013】

【発明の実施の形態】図1に、本発明によるデータ処理システムの好適な実現例を示す。アプリケーション1（102）を実行するクライアント1（101）、アプリケーション2（104）を実行するクライアント2（103）はネットワーク121を介してデータ処理システム105に接続される。ネットワーク121は、イーサネット、光ファイバ、FDDIで接続されるローカルエリアネットワーク（LAN）、もしくはLANよりも低速なインターネットを含んだワイドエリアネットワーク（WAN）でも差し支えない。クライアントは（株）日立製作所のHitachi FLORAなどのパーソナルコンピュータ、（株）日立製作所のHitachi 3500ワークステーションなどの任意のコンピュータシステムでよい。データ処理システムは、ネットワーク122を介して実データベースを管理するサーバ（126、127）に接続されている。ネットワーク122はネットワーク121と同様に、LAN、もしくはWANで差し支えなく、サーバはその上でデータベースマネジメントシステム（DBMS）（123、124）が稼働する任意の計算機で差し支えない。DBMSは（株）日立製作所のデータベースシステムHiRDBや、Oracle社のOracle8、IBM社のDB2などの汎用のDBMS製品でよい。

【0014】今アプリケーション2がデータ処理システ

ム上の仮想表119上の項目を参照する問合せ処理要求126を発行したとする。本実施例における仮想表は複数の実データベースを論理的に統合した表であり、仮想表上の項目は実データベース上の項目に多重マッピングされている。例えば、図2は貯蓄平均表201、および顧客貯蓄残高表202という異なる2つの実データベース上に存在する、2つの表が仮想表203に多重マッピングされている例である。仮想表の項目である支店(208)、および貯蓄平均(209)は、貯蓄平均表の支店(203)と貯蓄平均(204)、および顧客貯蓄残高表202の支店(206)と、該顧客貯蓄残高表の1億円以上の高額預金者を除いた顧客の預金すなわち貯蓄額(207)を支店別に集計し、平均を取った値の2通りにマッピングしている。貯蓄平均表の項目は仮想表の項目そのものにマッピングされているのに対し、顧客貯蓄残高表では集計演算の結果が割り当てられているのが大きな相違点であり、本マッピングによる仮想表定義は例えば210に示すような構文で記述可能である。マッピング記述例では仮想表Vの項目である支店と貯蓄平均に対して、貯蓄平均表の項目をマッピングしたVMA P1と、顧客貯蓄残高表から得た集計値をマッピングしたVMA P2が定義されている。

【0015】多重定義されたマッピングは、条件に応じて切り替えて利用される。切り替え指定方法の例を図3に示す。マッピング切り替え指定方法301に切り替え指定条件の例を5つ挙げているが、これらを順に説明する。まず、マッピング切り替え指定条件にデータ鮮度302を用いるのは、クライアント上のアプリケーションが必要とするデータの鮮度が異なる場合で、例えば図2で、顧客貯蓄残高表は顧客の取引が行われる毎に常に最新に保たれているのに対し、貯蓄平均表は1日単位で集計が行われて作成されているとする。あるアプリケーションは常に最新データを必要とするのであれば、顧客貯蓄残高表から要求がある毎に集計値を計算しなおす必要があるが、この場合にはVMA P2を用いる必要がある。これに対して、あるアプリケーションでは1日前の集計値で十分であるという要求であれば、VMA P1を用いて貯蓄平均表にアクセスすれば、集計が既に行われているデータにアクセスを行うため結果が速く得られ、しかも集計を行うシステム負荷を削減することができる。次に、データ精度303を用いるのは、例えばサンプリングなどの手法の適用によって、精度が異なるデータベースが複数存在する場合に、アプリケーションからのデータ精度の要求に応じてアクセスを切り替える場合である。ユーザロール304を用いるのは、例えば企業の意思決定を行うエグゼクティブはシステムに負荷をかけても最新データをアクセスする必要があるが、顧客対応で営業端末から検索を行う担当者レベルでは最新ではなくても素早くデータを取得する必要があるなどのユーザロールに応じたマッピングの切り替えを行いたい場合で、

図2の例ではエグゼクティブからのアクセスにはVMA P2を、担当者レベルからのアクセスにはVMA P1を用いる等である。業務によるマッピング切り替えは前記ユーザロールによる切り替えと似ているが、ユーザロールがユーザ個人に対する切り替えであったのに対して、窓口業務にはVMA P1を、企業方針決定シミュレーション業務にはVMA P2など、業務単位でマッピングを切り替える指定方法である。最後にアクセス位置306を用いるのは、図2の例で貯蓄平均表が東京センターに、顧客貯蓄残高表が大阪センターにある場合で、アプリケーションを実行する計算機が東京にある場合に、地理的にデータに近い東京センターのデータを利用するようにVMA P1を指定する方法である。マッピングの指定は例えば307に示すような構文で行う。307はデータ鮮度の要求が1日より小さい場合、すなわち1日毎の集計値では満足できない場合にはVMA P2を、そしてデフォルトではVMA P1を使う例である。ここではそれぞれのマッピング切り替え指定を個別に指定する例を説明したが、これらを組合せて利用しても差し支えない。

【0016】図1に戻って、データ処理システム105内のアプリケーションマネージャ106はアプリケーション2から発行された問合せをネットワーク経由のクライアントからの問合せ処理要求を受け付ける。受け付けられた問合せは問合せ解析部107で解析され、問合せ最適化部108に転送される。データ処理システムにおける問合せ処理の流れを図1および図11および図13を用いて説明する。問合せ最適化部では、まず受け付けた問合せを図13の1301に示すような選言標準形(disjunctive normal form)に変形する(1102)。ただし、図13の1305の構成要素となっている各qは原子論理式を表す。任意の論理式は該標準形へ変換できることが保証されており、変換方法は例えば、Chin-Liang Chang、Richard Char-Tung Lee著、長尾、辻井訳、“コンピュータによる定理の証明”の第2章4節(文献7)に示されている。選言標準形に分解され、外側がORで結合された内部のANDで結合された部分1302を以下では部分問合せと呼ぶ。

【0017】データ処理システム内の問合せ最適化部108では問合せを選言標準形に変形した後、部分レプリカ管理部113で管理される部分レプリカ情報を利用して、部分問合せ単位でデータ処理システム内のストレージ120に格納された部分レプリカ125を利用できるか否かを判定する。ストレージ120は磁気記憶装置、フラッシュメモリ、もしくはメモリで構わない。また部分レプリカ管理部は、部分レプリカをファイルとして管理するのでもデータベースシステムを用いてデータベースとして管理するのでも差し支えない。

【0018】仮想表と部分レプリカの関係を図4を用い

て説明する。401は仮想表定義で指定される範囲を表す。部分レプリカは仮想表の一部を実体化したデータであり、本例においては領域417および418が、部分レプリカが作成されている範囲を示しており、該範囲のデータはストレージ411内に部分レプリカ412として格納される。部分レプリカは仮想表に対して定義され、例えば、部分レプリカ定義例422のような構文で記述できる。これは、図2の仮想表定義VMAP1(201)の貯蓄平均が20万円より大きい範囲のデータの部分レプリカをPR1という名前で作成する例である。

【0019】部分レプリカを用いて仮想表に対する問合せを全て処理することができない場合には、部分レプリカと実データベースのデータを組合せて処理する。例えば、クライアント402上のアプリケーション403が発行した問合せ404を処理するために、破線で囲まれた領域406のデータが必要である場合、領域406は部分レプリカ417を用いて処理できる領域408と、実データベース416を用いて処理する必要がある領域407に分けられる。

【0020】部分レプリカ管理部は、部分レプリカとしてデータを保持すると共に、該部分レプリカがどの範囲のデータのコピーであるかを管理する部分レプリカ管理テーブル601を保持する。図6に該部分レプリカ管理テーブルの例を示す。部分レプリカ管理テーブルは、部分レプリカID(602)、部分レプリカ名(603)、部分レプリカを格納するテーブルの項目名であるレプリカ項目名(604)、レプリカ作成対象の仮想表名(605)、前記レプリカ項目名に対応する仮想表上の項目名、もしくは仮想表上の項目に対する演算結果であるレプリカ作成対象(606)、レプリカ作成対象の仮想表のマッピング名(607)、そしてレプリカ作成条件(608)を項目として含み、問合せ処理の部分レプリカ利用判定に利用される。例えば、部分レプリカ管理テーブルの部分レプリカIDが1のエントリは、図4に示した部分レプリカ定義422に対応する。該部分レプリカ管理テーブルは、部分レプリカのデータと同様にストレージ120に格納して差し支えない。

【0021】部分問合せQi(1302)が部分レプリカを用いて処理可能か否かは、部分問合せを構成する全ての原子論理式1305が部分レプリカを用いて処理可能かどうかで判定する。具体的には、原子論理式と同値もしくは該原子論理式を包含する部分レプリカが存在する場合に、該原子論理式は該部分レプリカで処理できる。部分レプリカの情報は前述の部分レプリカ管理テーブル601を参照すればよい。部分レプリカ定義が該部分レプリカ管理テーブルの部分レプリカIDが1のエントリである場合を考える。問合せが該部分レプリカ定義の一部を参照する条件である場合、例えばSELECT 支店 FROM VT WHERE 貯蓄平均>300,000;の場合には、該問合せは該部分レプリカを

用いて処理できる。処理可能か否かの判定は、本例の場合以下のように行う。但し、以下の説明では前記問合せ例では“支店”に相当するSELECT句の選択の対象となる項を取得対象、該問合せでは“VT”に相当するFROM句に含まれる仮想表名を取得対象仮想表名、そして該問合せでは“貯蓄平均>300,000”に相当するWHERE句に指定されるデータ取得のための条件を取得条件と呼ぶ。

【0022】最初に問合せ文を解析し、取得対象、取得対象仮想表名、取得条件を取り出す。本例の場合には前記取得対象、前記取得対象仮想表名、前記取得条件はそれぞれ“支店”、“VT”、“貯蓄平均>300,000”となる。次に該取得対象と図6に示した前記部分レプリカ管理テーブルのレプリカ作成対象606を比較する。本例の場合には取得対象がレプリカ作成対象に含まれているため、取得対象項目に関するチェックはOKとなる。次に取得対象仮想表名と前記部分レプリカ管理テーブルの仮想表名(605)を比較する。本例の場合には、取得対象仮想表名と該部分レプリカ管理テーブルの仮想表名が等しいので、取得対象仮想表名に関するチェックもOKとなる。最後に前記取得条件とレプリカ作成条件(607)を比較する。本例では該取得条件が該レプリカ作成条件に包含されるため、レプリカ作成条件に関するチェックもOKとなり、該問合せが部分レプリカPR1で処理できると判定される。ここではレプリカ定義、問合せが共に非常に単純な場合を想定したが、例えばJeffrey D. Ullman著、“PRINCIPLES OF DATABASE AND KNOWLEDGE-BASE SYSTEMS”、VOLUME II、COMPUTER SCIENCE PRESS、ISBN 0-7167-8162-X、第14章“Optimization for Conjunctive Queries”(文献8)に示されている、Query Equivalence、Query Containmentと呼ばれる方式を用いることにより、一般的な場合についても条件間の同値関係、包含関係を調べる方法が開示されており、該判定方式を利用することにより、一般的な場合についても本発明による部分レプリカ利用判定処理が実現できる。

【0023】問合せ最適化部で、全ての部分問合せが部分レプリカを用いて処理可能であると判定された場合、すなわちステップ1103でYesが選択された場合には、問合せは部分レプリカに対して適用される(1107)。この場合、問合せは問合せ生成部109で部分レプリカを参照するように書き換えられ、問合せ実行部110がローカルデータベースアクセス部116を介してストレージ120に格納されている部分レプリカ125を用いて問合せ処理を行い、得られたデータを結果127としてアプリケーションマネージャ経由でクライアントに返して問合せ処理を終了する(1108)。問合せ

最適化部で、全ての部分問合せは部分レプリカを用いて処理できないと判定された場合、すなわちステップ1103でNoが選択された場合には、部分問合せのうち少なくとも一つが部分レプリカを用いて処理可能かどうかをチェックする(1104)。一つの部分問合せも部分レプリカを用いて処理できない場合、すなわちステップ1104でNoが選択された場合には、問合せ全体を実データベースで処理する必要があるため、問合せ生成部で該部分問合せを実データベースを参照する問合せに書き換え、問合せ実行部が必要に応じて問合せの構文を構文変換部115を用いて変換し、リモートデータベースアクセス部111を介して実データベース(117、118)に対して問合せ処理を実行する(1105)。実データベースから得られたデータは、必要に応じてデータ型、値変換部114でデータをクライアントの要求する形式に変換し、結果127としてクライアントに返して問合せ処理を終了する(1108)。問合せ最適化部で部分問合せの少なくとも一つが部分レプリカを用いて処理できると判定された場合、すなわちステップ1104でYesが選択された場合には、問合せ生成部は部分レプリカを用いて、処理可能な部分問合せは部分問合せを参照する問合せに書き換え、問合せ実行部がローカルデータベースアクセス部を介して部分レプリカを用いて問合せ処理を行い、それ以外の部分問合せに関しては問合せ生成部で実データベースを参照する問合せに書き換え、問合せ実行部が必要に応じて問合せの構文を構文変換部を用いて変換し、リモートデータベースアクセス部111を介して実データベースに対して問合せ処理を実行し、実データベースから得られたデータは、必要に応じてデータ型、値変換部114でデータをクライアントの要求する形式に変換し、該問合せ実行部で前記部分レプリカから得られたデータと前記実データベースより得られた結果を統合し、問合せ126に対する結果127としてクライアントに返して問合せ処理を終了する。図5は前述の図4を用いて説明した例のように、アプリケーションからの仮想表を介した問合せが、部分レプリカと実データベースの双方のデータを必要とする場合の結果統合の模式図である。アプリケーション503からの問合せ504が部分レプリカ511および実データベース2(516)の双方のデータを必要とする場合には、部分レプリカから得られた結果507と実データベース516から得られた結果508とを問合せ実行部501で統合し、問合せ処理の結果505としてアプリケーション503に返す。

【0024】問合せ最適化部で部分レプリカの使用可否を判定した後、選言標準形に変形された問合せは問合せ生成部109に転送される。問合せ生成部では、該問合せを前述したように部分レプリカを参照する問合せと実データベースを参照する問合せに書き替える。部分レプリカを参照する問合せに関してはローカルに書き換えた

問合せをストレージ120に転送し、結果を得ることができるが、図8に示すように、クライアント801上のアプリケーション802からの仮想表に対する問合せ803がネットワーク807を介した異なる実データベース810、812間の結合処理808を必要とする場合である。この場合にはデータベース1(810)およびデータベース2(812)から取得するデータ量やネットワーク(807)の帯域幅、および結合結果の大きさの見積り等から、最適な実行方式を決定する必要がある、処理ステップは複雑になる。そこで、図1および図12および図13を用いて具体的なステップを説明する。

【0025】実データベースに対する問合せは問合せ生成部で連言標準形(conjunctive normal form)に変形される(1202)。連言標準形とは図13の1303に示されるように、外側がAND、内側がORで結合される形であり、選言標準形と同様に任意の論理式は連言標準形に変形できる。変換方法は選言標準形の場合と同様に、文献7の第2章4節に示されている。連言標準形の内部のORで結合された部分1304を選言問合せと呼ぶ。選言問合せが一つの実データベース(117)に対する問合せである場合、すなわちステップ1203でYesが選択された場合には、該問合せは該実データベース内のデータのみを用いて処理できるため、問合せ生成部109は該選言問合せ処理を該実データベース117で実行するように処理をブッシュダウンする問合せ書き替えを行う。問合せのブッシュダウンは、データ転送量を削減するために効果が大きい。例えば、図7に示すような仮想表と実データベースが存在する環境を仮定する。図7に示す環境では、725に示される仮想表定義でサーバ1(711)に接続されているデータベース1(715)内のテーブルT1(723)と、サーバ1とは異なるサーバ2(712)に接続されているデータベース2(716)内のテーブルT2(724)の項目が仮想表702にマッピングされている。この時、図9に示した問合せ1(901)が投入されたとすると、2つの条件 $ic2 \leq 10$ と $ic4 \geq 10$ はそれぞれデータベース1、データベース2にブッシュダウンでき、各々のデータベースのデータをデータ処理システムに転送してから該2つの条件を適用するよりもネットワーク710を転送されるデータ量は遥かに小さくなる。

【0026】問合せ1と同様に問合せ3(903)も本発明の問合せ最適化が有効な例である。問合せ3は選言標準形のままでは条件をブッシュダウン可能かどうかは判定できないが、該問合せを連言標準形に変形すると904のようになる。まず、選言問合せ905は $ic2$ だけを参照する条件であり、仮想表定義725によれば $ic2$ は $c2$ にマッピングされるので、該選言問合せはデータベース1(715)にブッシュダウンできる。ま

た、選言問合せ908はic4だけを参照する条件であり、前記仮想表定義によればic4はc5にマッピングされるため、該選言問合せはデータベース2にブッシュダウンされる。選言問合せ906および907はデータベース1もしくはデータベース2にブッシュダウンできないが、問合せが連言標準形に変換されているため、前記ブッシュダウンされた905および908の適用によって削減されたデータに対してのみ、データ処理システム内で選言問合せ906および907を適用すればよく、データ転送量削減の効果が得られる。

【0027】選言問合せが一つの実データベースに対する問合せでは無い場合、すなわちステップ1203でNoが選択された場合には該選言問合せは一つの実データベースだけでは処理できない。この場合、問合せ生成部では該選言問合せ、もしくは仮想表定義が異なるデータベース間の結合処理を含むかどうかをチェックする(1204)。ステップ1204でNoが選択された場合には、結合処理が含まれない場合であるので、問合せ生成部は実データベースからデータをデータ処理システムに転送し、その後条件を適用する問合せを生成し、問合せ実行部は実データベースからデータを転送して必要に応じて変換し、データ処理システム上のストレージを用いて収集したデータに条件を適用して結果を生成し、実データベースに対する問合せ処理を終了する(1210)。

【0028】最後に、選言問合せが異なるデータベース間の結合処理を含む場合、すなわちステップ1204でYesが選択された場合には、結合処理に関しては分散ジョイン方式を適用し、ジョイン後にデータ処理システムで残りの条件を適用して結果を生成し、実データベースに対する問合せ処理を終了する。例えば、図9の問合せ2(902)は、WHERE句の条件であるic2<ic5はデータベース1のみでもデータベース2のみでも評価できず、結合処理でもないが、仮想表定義725にデータベース1上のテーブルT1とデータベース2上のテーブルT2の結合処理が存在するため、T1とT2の結合処理に分散ジョインを適用した後、データ処理システム内で条件ic2<ic5を適用する。

【0029】分散ジョインを行う必要があるのは、図8に示した例のように、異なるデータベース上にあるデータの結合処理が必要な場合である。以下の説明では図8のデータベース1(810)上のテーブルT1(815)と、相異なるデータベース2(812)上のテーブルT2(816)を808で示される条件で結合(分散ジョイン)することを想定する。本発明のデータ処理システムでは分散ジョインの実行方式として以下の4方式を備え、条件に応じて問合せ生成部が該4方式のうちの1つを選択し、問合せ実行部が該選択方式を実行する。

【0030】第1の方式は結合対象のデータをデータ処理システムに転送し、データ処理システムに接続されて

いるストレージに一時的に格納してデータ処理システムで結合処理を行う。図8の例では、T1でc2<10を満足するレコード全てを結合対象データとしてネットワーク807を介してデータ処理システムが受け取り、該結合対象データをストレージ806内の一時テーブルT3(817)に格納する。同様にT2でc4>100を満足するレコード全てを結合対象データとしてデータ処理システムに転送し、ストレージ内の一時テーブルT4(818)に格納し、前記T3とT4をデータ処理システムで結合する。本方式をストレージ利用ジョインと呼ぶ。第2の方式は、第1のデータベースから得た結果を第2のデータベース検索の問合せに埋め込むことによって結合を実行する方式である。図8の例では、第1のデータベースに対してSELECT c2 FROM T1 WHERE c2<10;問合せを発行した結果得られたc2の値が{2, 8}であったとすると、データ処理システムではデータベース2に対して、SELECT c3, c4 FROM T2 WHERE (c3=2 OR c3=8) AND c4>100;もしくはSELECT c3, c4 FROM T2 WHERE c3 in (2, 8) AND c4>100;という問合せを生成し、該問合せをデータベース2に対して発行して得た結果を結合結果とする。本方式をORジョインと呼ぶ。第3の方式は、第2の方式と同様にまず第1のデータベースから結果を得、第2のデータベースに対する検索式には条件をパラメータとして記述し、第1のデータベースから得た結果をパラメータの値として与えながら結合を実行する。図8の例では、第2の方式と同一の問合せを第1のデータベースに発行し、得られた結果も同一の{2, 8}であったとする。第3の方式では第2のデータベースに対して、パラメータ:xを含む問合せSELECT c3, c4 FROM T2 WHERE c3=:x AND c4>100;という問合せを発行し、パラメータ:xの値として{2, 8}を与えて結果を得る。本方式をパラメータ埋め込みジョインと呼ぶ。最後に第4の方式は、第1のデータベースから得た値を第2のデータベースの一時表に挿入し、第2のデータベース上で結合を行う方式である。図8の例では、第2および第3の方式と同様に、第1のデータベースに対して発行した問合せの結果得られたc2の値{2, 8}をデータベース2上の一時表T5(819)に挿入し、データベース2でSELECT c3, c4 FROM T2, T5 WHERE c3=c2' AND c4>100;という問合せを実行し結果を得る。本方式を挿入ジョインと呼ぶ。

【0031】問合せ生成部が前記分散ジョインの4方式を選択する際には、適用可能性をチェックした後に適用可能なジョイン方式から処理コストが最も小さな方式を選択する。分散ジョイン方式選択処理を図10を用いて説明する。分散ジョイン方式適用可能性のチェック10

01においては、まず分散ジョイン実行方式リストに、前記4方式をセットする(1007)。次に、結合対照表を保持する第2のデータベースに一時表を作成可能かをチェックする(1002)。第2のデータベースに一時表を作成できない場合、すなわちステップ1002でNoが選択された場合には、分散ジョイン実行方式リストから前記挿入ジョイン方式を削除する(1003)。次に第1のデータベースから得られた結果を第2のデータベースへの問合せに埋め込んだ問合せ長が、第2のデータベースの問合せ言語の問合せ長制限を越えないかどうかをチェックする(1004)。例えばISOで規格化が行われ、殆ど全てのDBMSで問合せ言語として広く用いられているSQL(Structured Query Language)の各DBMSへの実装には通常SQL長制限があるため、第1のデータベースから得られる結果が多すぎる場合にはORもしくはINを用いて全ての結果をSQLに埋め込むことができない。このように問合せの長さが問合せ長制限を超える場合、すなわちステップ1004でYesが選択された場合には、分散ジョイン実行方式リストからORジョイン方式を削除する(1005)。そして最後に分散ジョイン実行方式リスト中の実行方式の処理コストを見積り、処理コストが最小となる実行方式を選択して(1008)、分散ジョイン方式選択処理を終了する(1006)。

【0032】ステップ1008の処理コストCは、データの転送に関するコストCtとデータベース内部処理のコストCdとの和、すなわち $C = C_d + C_t$ で算出する。Cの算出方法の概略を図8の結合処理808を例にして説明する。T1、T2から選択条件適用後に得られる結果のデータ量をそれぞれA1、A2、その行数をそれぞれL1、L2、ネットワーク807の帯域幅をWn、結合処理の結果のデータ量をA12、行数をL12、ストレージ806への1行あたりのデータ挿入コストCis、結合処理コストをCjs、サーバ2における1行あたりのデータ挿入コストをCi2、結合処理コストをCj2とする。この時、ストレージ利用ジョインのデータ転送コスト $C_t = A1/Wn + A2/Wn$ 、 $C_d = \text{ストレージ806での挿入コスト} + \text{結合処理コスト} = (L1 \times C_{is} + L2 \times C_{is}) + (L1 \times L2 \times C_{js})$ と近似できる。よって、ストレージ利用ジョインコストは $C = C_t + C_d = A1/Wn + A2/Wn + (L1 \times C_{is} + L2 \times C_{is}) + (L1 \times L2 \times C_{js})$ となり、該計算結果に環境依存のコスト値を代入することによって、具体的なコストの近似値を求めることができる。ORジョイン方式、パラメータ埋め込みジョイン、挿入ジョインの各方式についても同様の計算で近似コストが計算できるので、該計算値を比較することによって、最小コストとなる方式を決定できる。データ転送コストCtはより単純な方法としては、結果行数で近似

しても差し支えない。

【0033】該処理コストの計算において、各データベース内部処理方式が不明でその内部処理コストCdが求められない場合、もしくは内部処理コスト計算時間そのものを省略したい場合には、処理コストCを転送処理コストCtのみで近似する方法もある。これはデータ転送処理コストが内部処理コストよりも大きい場合には誤差が少なく、しかも計算処理時間を短縮する意味で有効な方法である。

10 【0034】上記の実施例では、一つの結合処理の選択方式について述べたが、問合せによっては複数の結合処理を含む場合がある。複数の結合処理を含む問合せ実行方法の選択方式を図17および図18を用いて説明する。今、データ処理システム1701内の仮想表1702に対して発行された問合せが1721に示すような実データベースに対する問合せに変換されたとする。該問合せは二つの結合処理を含むため、結合処理の順序を考慮した問合せ実行方法を木の形で表すと図18の(a)と(b)に示す2種類がある。例えば、図18(a)は、図17のデータベース1(1710)上にあるテーブルT1(1712)とデータベース2(1711)上にあるテーブルT3(1714)を、条件T1.c1=T3.c5で結合(1804)し、その後その結果とデータベース1上にあるテーブルT2を条件T1.c2=T2.c3で結合(1803)することを表す。

20 【0035】問合せ実行方法の選択方式の第1ステップでは、図18に示すように木の形の異なる実行方法の候補を数え上げる。本実施例の問合せ(1721)では、考えられる木の形は図18に示す2通りである。それぞれの結合処理の実行方式は前述の分散ジョイン処理選択方式で説明したように4種の組合せが考えられるため、例えば図18(a)では単純計算で結合処理1803および1804の結合方式の組合せは $4 \times 4 = 16$ 通りとなる。さらに、該分散ジョイン処理選択方式で説明したように、結合処理を実行する際にはデータを動かす方法によって、結合対象の2つのテーブル(もしくは中間結果)を第1の表と第2の表とするかも考慮する必要があるため、全体として64通りの組合せを考慮する必要がある。

40 【0036】次に第2ステップでは、第1ステップで数え上げた実行方法の候補中から、分散ジョイン方式選択のステップ中にあったように実行が不可能なジョイン方式を削除する。例えば、図18のデータベース2(1711)には挿入権限がない場合には、結合処理1804でT3を第2の表とした挿入ジョイン方式を候補から削除する。

50 【0037】第3ステップではまず各実行方法の候補中の各結合処理コストを計算する。分散ジョイン処理のコストの計算方法で説明した通り、処理コストCをデータ転送処理コストCtとデータベース内部処理コストCd

21

の和 $C_t + C_d$ で計算しても、処理コスト C をデータ転送処理コスト C_t で近似しても差し支えない。そして各結合処理方法のコストの和を計算して、該コストを実行方法の候補のコストとする。データ転送量を処理コストとみなして処理コストを計算する場合の例を図18を用いて説明する。図18(a)の結合処理1804で、挿入ジョイン方式を選択する場合のコストを計算することを考えた場合、図17のT1を第1の表、T3を第2の表とする場合には、データベース1上のT1のレコードをデータベース2上の一時表に挿入し、データベース2上で結合処理を実行することとなる。この場合には、その次の結合処理1803でさらにデータ転送処理が発生するのに対して、T3を第1の表、T1を第2の表として前記挿入ジョイン方式を実行すると、一時表はデータベース1上に生成されるため、結合処理1803ではデータベース間でのデータ転送処理は発生せず、後者のコストは前者のコストよりも小さくなると推定できる。

【0038】最後に第4ステップでは、第3ステップで計算した各実行処理方法コストを比較し、そのコストが最も小さい実行処理方法候補を実行処理方法として選択する。以上が結合処理を複数含む場合の間合せ実行方法の選択方式の具体的な説明である。

【0039】最後に図1の構文変換部115、およびデータ型、値変換部114について説明する。構文変換部はDBMS毎に間合せ構文が異なる場合に必要となる構文変換を実現する。具体的には構文変換部はデータ処理システムで規定する標準構文と、各DBMSの構文との対応表を保持し、変換が必要な構文は該対応表を用いて置き換えを行い、各DBMS向けに変換する。次にデータ型の変換に関しても構文変換部と同様に、データ処理システムのデータ型と各DBMSのデータ型の対応表を管理し、実データベースからデータ処理システムにデータが転送される際に前記対応表を参照して適切な型変換プログラムモジュールを起動し型変換を実行する。図14に前記対応表の例を示す。型変換対応表は、DBMS名称(1402)、DBMSのバージョン(1403)、該DBMSでのデータ型名称(1404)、該データ型に対応するデータ処理システムのデータ型(1405)、変換元DBMSのデータをデータ処理システム側に交換する場合に使用する、正変換プラグインモジュール(1406)、そしてデータ処理システム側のデータを前記DBMS側に交換する場合に使用する、逆変換プラグインモジュール(1407)のエントリを含む。例えばエントリ1408は、DBMS_Aのバージョン5.0のINT型のデータをデータ処理システム側のINTEGER型に対応付けるために、正変換は`Int_to_Integer()`、逆変換は`Integer_to_Int()`というプラグインを用いることを表している。図1の間合せ実行部110は該プラグインを用いることによって、データ変換を実現する。

22

【0040】エントリ1408のように、DBMS_Aのデータ型INTに対してデータ処理システム側に対応する適当なデータ型INTEGERが存在する場合には、データ変換は可能であるが、エントリ1409はDBMS_BのNUMBERというデータ型に対応するデータ型がデータ処理システム側に存在しない場合である。NUMBERが巨大な整数で、データ処理システム側には該整数を適切に取り扱えなう型が存在しない場合などが好適な例である。このような場合にも適切にデータを取り扱えるように、本発明のデータ処理システムでは普遍データ型(Universal Data Type)1501というデータ型を備える。該普遍データ型は図15に示すように、DBMS名称タグ(例えば2バイト)1502、バージョンタグ(例えば1バイト)1503、データ型名称タグ(例えば2バイト)1504で構成されるデータ型識別子1506と、データ本体を格納するBLOB(BINARY LARGE OBJECT)1505で構成されるデータ型である。BLOBデータ型は(株)日立製作所製のDBMSであるHiRDBをはじめ、多くのDBMSでサポートされている。先の例における巨大整数NUMBERも、該普遍データ型を用いることにより、データ処理システムでの格納および処理が可能となる。

【0041】型、値変換部における値の変換とは、例えば図7の結合処理においてc3、c4が共に整数値で性別を表している項目であっても、c3では男性が0、女性が1であるのに対して、c4では逆に女性が0、男性が1であるように値の意味的な変換が必要となる場合に行う処理である。型変換の場合と同様に、値変換の場合も図16に示す値変換対応表1601を値変換部が管理し、値変換プラグインモジュールを値変換時に適用することによって実現する。例えば、図7の例では、仮想表VT1上の項目ic3では男性が1、女性が0で表されており、c3とは逆であるので、値の変換のためにプラグインモジュール`InverseInt()`を用いることである。

【0042】以上の説明において、データ処理システムは一つであるとしていたが、複数のデータ処理システムがネットワークで接続され、相互に情報を交換して動作する場合にも本発明が有効であることはいうまでもない。また、以上の説明においては、アクセス対象の項目を関係データベースのテーブル上の項目としており、あたかも関係データベースを前提としているように述べたが、アクセス先の項目がオブジェクトであるオブジェクト指向データベースでも、タグ付けされた半構造データであっても差し支えない。さらに、本説明においては、間合せ及び部分レプリカの定義をSQL形式で記述していたが、間合せ言語及び部分レプリカ記述言語は同等の能力を持つ任意の言語で差し支えなく、言語を用いずにツールを用いて内部情報をセットする実装にも本発明を

利用できることはいうまでもない。

【0043】なお、第10図～第12図の処理手順をコンピュータで実行するためのコンピュータプログラムを例えば可搬型の記録媒体に記録しておけば、本発明の方法を任意の場所で行うことができることになる。

【0044】

【発明の効果】本発明のデータ処理システムを用いることにより、仮想表上のアクセスする項目の変更、もしくは仮想表の切替え無しに、条件に応じて複数の異なる実データベースに透過アクセスすることができ、異なる業務開発毎に新アプリケーションを作成する必要がなくなる。これにより、アプリケーションの開発コスト、維持コストを削減できると共に、アプリケーション開発時間の削減による新規業務への迅速な取り組みが実現できる。

【0045】さらに、本発明のデータ処理システムを用いることにより、部分レプリカを利用した高性能の分散問合せ処理を実現することができ、大規模な複数のデータベースに対する負荷の高い複雑な問合せを短時間で処理することができ、企業の意思決定支援を迅速に行うことが可能となる。

【図面の簡単な説明】

【図1】本発明におけるデータ処理システムの構成を示す図。

【図2】本発明における多重マッピング可能な仮想表定義例を示す図。

【図3】本発明における仮想表のマッピング切替え指定方法を示す図。

【図4】本発明における仮想表経由の部分レプリカおよび実データベースアクセスを示す図。

【図5】本発明における実データベースと部分レプリカのデータの統合概要を示す図。

【図6】本発明における部分レプリカ管理テーブルを示す図。

【図7】本発明における結合処理を含む仮想表定義例を示す図。

【図8】本発明における結合処理方式説明のための図。

【図9】本発明における問合せ最適化説明のための問合せ例を示す図。

【図10】本発明における分散ジョイン方式選択処理のステップを示すフローチャート。

【図11】本発明におけるデータ処理システムでの問合せ

処理ステップを示すフローチャート。

【図12】本発明におけるデータ処理システムの実データベースに対する問合せ処理ステップを示すフローチャート。

【図13】本発明における問合せ変換で利用する問合せ標準形を示す図。

【図14】本発明における型変換対応表を示す図。

【図15】本発明における普遍データ型を示す図。

【図16】本発明における値変換対応表を示す図。

【図17】本発明における問合せ実行方法選択方式を説明する例を示す図。

【図18】本発明における問合せ実行方法の候補を示す図。

【符号の説明】

101、103、402、502、801…クライアント、

126、127、420、421、517、518、711、712、813、814、1706、1707…サーバ、

102、104、403、503、802…アプリケーション、

121、122、419、506、710、807、1705…ネットワーク、

123、124、413、415、513、514、713、714、809、811、1708、1709…DBMS、

117、118、414、416、515、516、715、716、810、812、1710、1711…データベース、

120、411、510、708、806、1703…ストレージ、

125、412、511、709、1704…部分レプリカ、

723、724、815、816、817、818、819、1712、1713、1714…テーブル、

417、418…部分レプリカ作成領域、

406…問合せ処理に必要な領域、

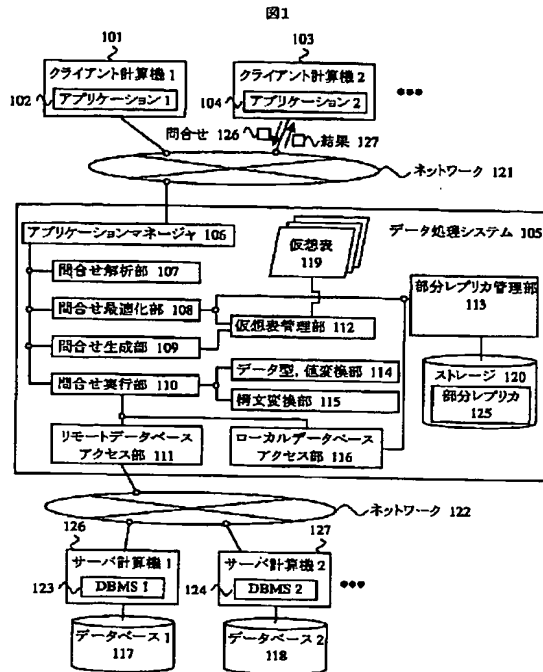
407…実データベースでの処理が必要な領域、

408…部分レプリカでの処理が可能な領域、

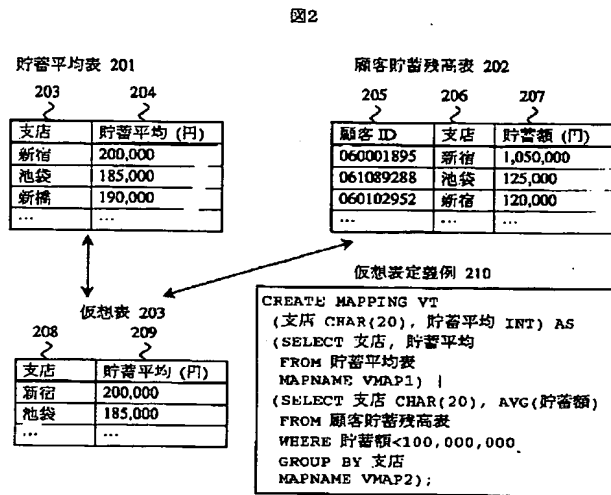
105、701、805、1701…データ処理システム、

119、203、401、702、1702…仮想表。

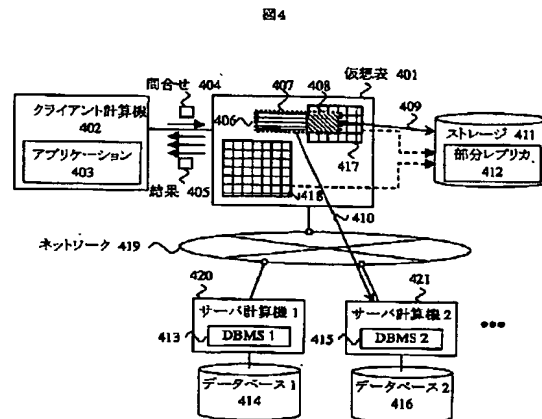
【図 1】



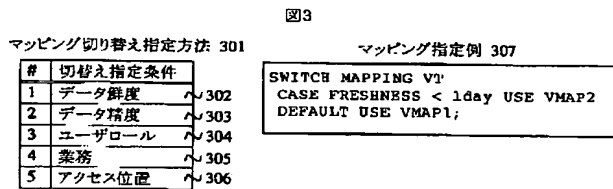
【図 2】



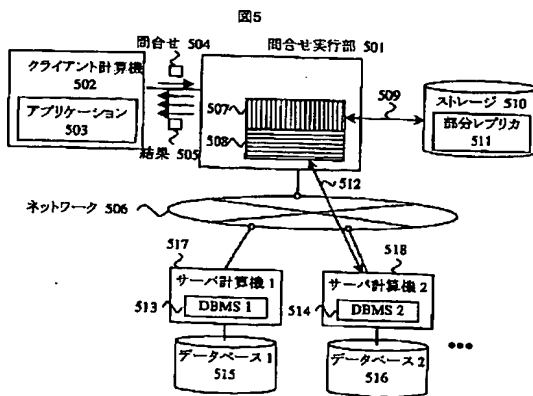
【図 4】



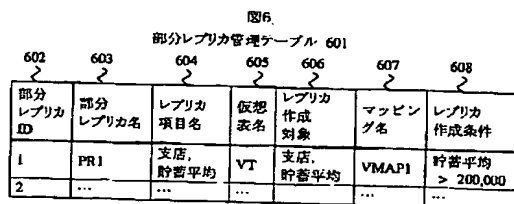
【図 3】



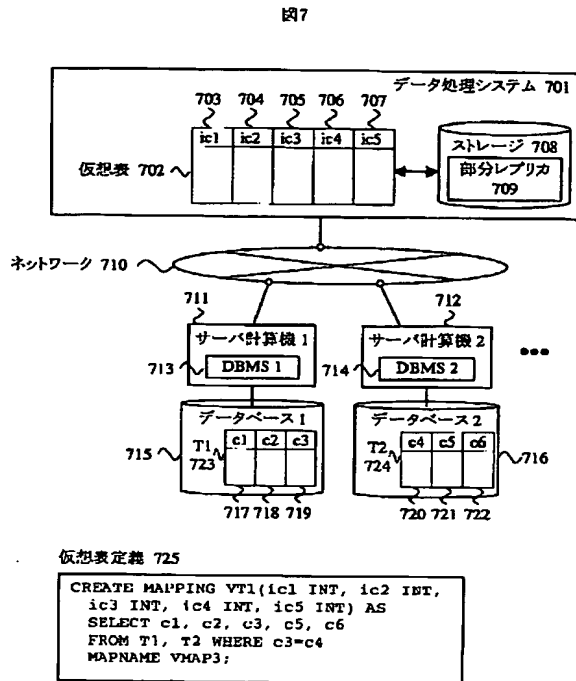
【図 5】



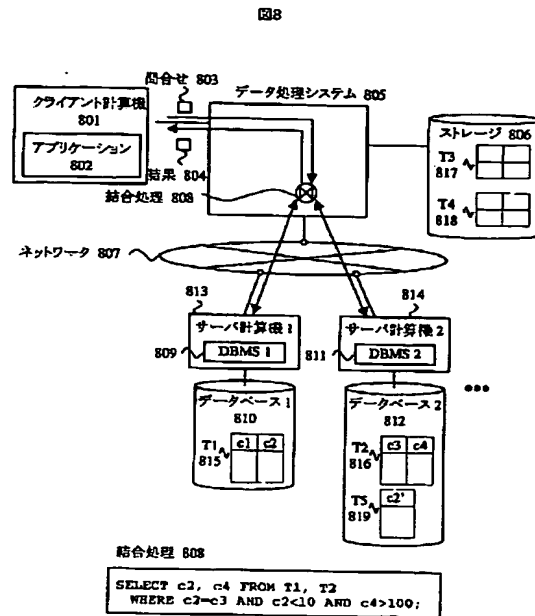
【図 6】



【図7】

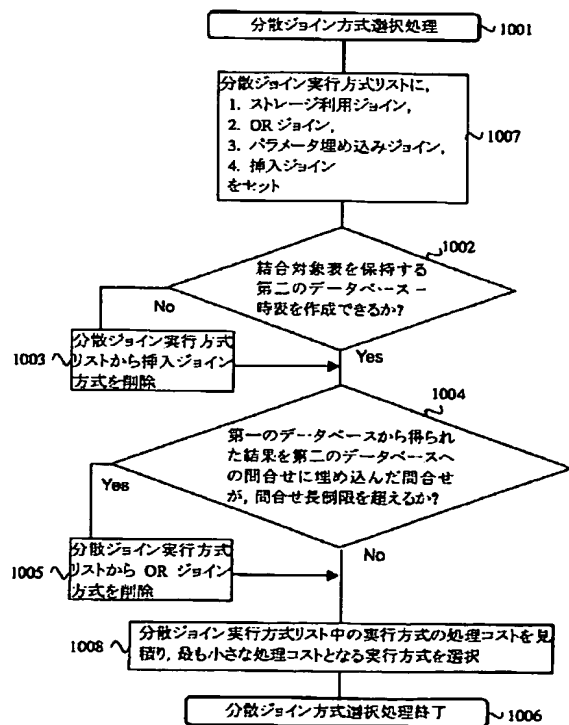


【図8】



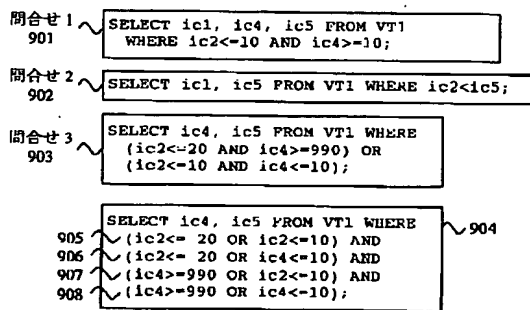
【図10】

図10



【図9】

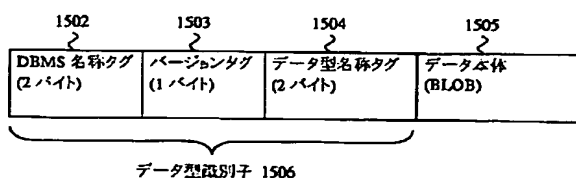
図9



【図15】

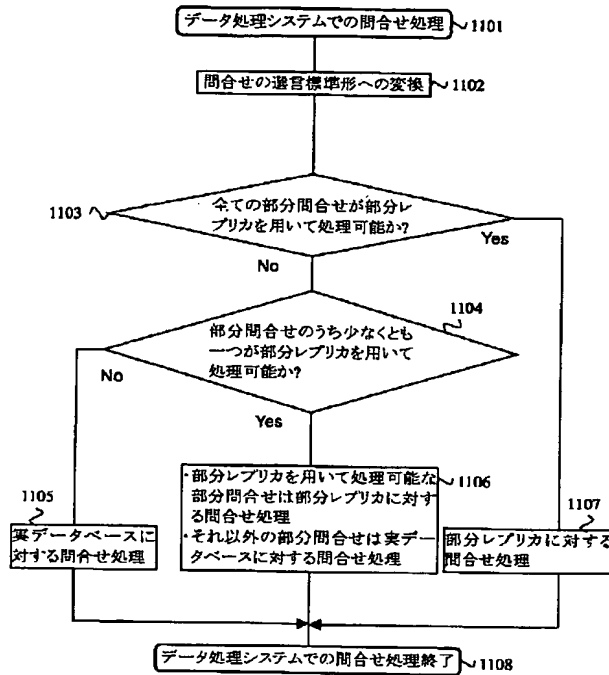
図15

普通データ型 1501



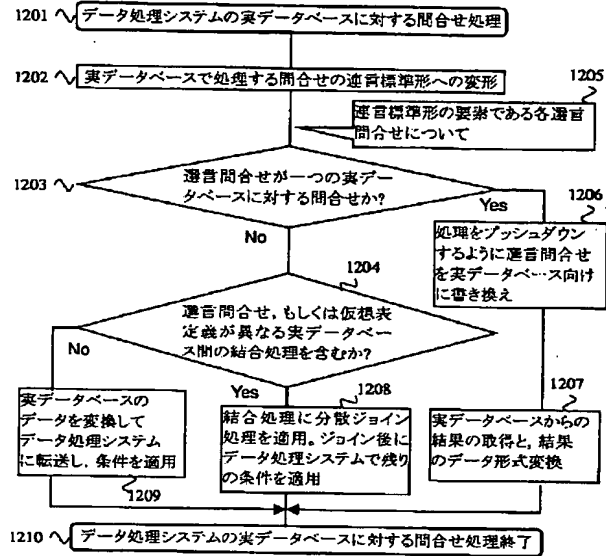
【図11】

図11



【図12】

図12



【図13】

図13

連言標準形 (disjunctive normal form)

$$1301 \sim Q_1 \sim Q_2 \text{ OR } Q_2 \text{ OR } \dots \text{ OR } Q_n$$

$$\text{但し, } Q_i = q_{i1} \text{ AND } q_{i2} \text{ AND } \dots \text{ AND } q_{in_i}$$

連言標準形 (conjunctive normal form)

$$1303 \sim Q_1 = Q_1 \text{ AND } Q_2 \text{ AND } \dots \text{ AND } Q_n$$

$$\text{但し, } Q_i = q_{i1} \text{ OR } q_{i2} \text{ OR } \dots \text{ OR } q_{in_i}$$

【図14】

図14

型変換対応表 1401

1402	1403	1404	1405	1406	1407
DBMS 名称	バー ジョン	データ型 名称	データ処 理システム 対応デー タ型	正変換 プラグイン モジュール	逆変換 プラグイン モジュール
1408 DBMS_A	5.0	INT	INTEGER	Int_to_Integ er()	Integer_to_In t()
1409 DBMS_B	8.0	NUMBER	UDT	Number_to_ UDT()	UDT_to_Nu mber()
...

【図16】

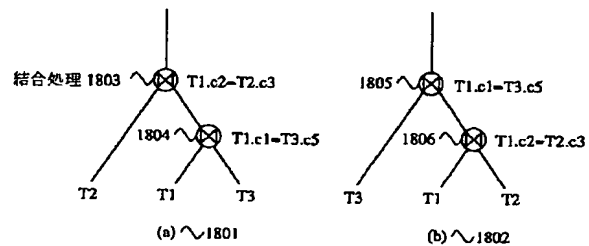
図16

値変換対応表 1601

1602	1603	1604	1605	1606	1607
仮想表名	仮想表 項目名	仮想表 マッピング名	実デー タベー ス項目名	正変換 プラグイン モジュール	逆変換 プラグイン モジュール
1608 VT1	ic3	VMAP3	c3	InverseInt()	InverseInt()
...

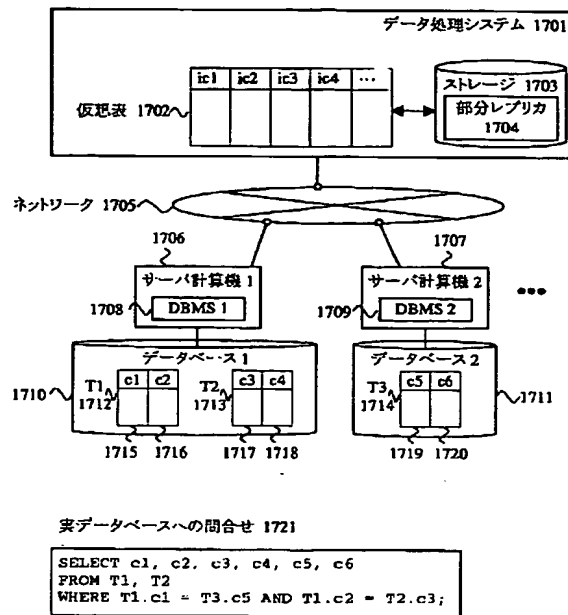
【図18】

図18



【図17】

図17



フロントページの続き

(72)発明者 佐川 暢俊
東京都国分寺市東恋ヶ窪一丁目280番地
株式会社日立製作所中央研究所内

(72)発明者 清水 晃
東京都国分寺市東恋ヶ窪一丁目280番地
株式会社日立製作所中央研究所内
F ターム(参考) 5B075 KK03 KK07 QT06
5B082 GA08 GC04
5E501 AC23 AC25 AC33 BA05

This Page Blank (uspto)